収束の加速法(数値計算アルゴリズムの現状と展望)

AUTHOR(S):

長田, 直樹

数理解析研究所講究録
第 880 巻 1994 年 28-43

収束の加速法

長崎総合科学大学 長田直樹 (Naoki Osada)

1. はじめに

自然科学や工学の問題を数学的に扱うとき，収束する数列やベクトル列がしばしば登場する．たとえば，非線型方程式を Newton-Raphson 法などの反復法で解くときなどである．単独方程式のときは数列，連立方程式のときはベクトル列が得られる．これらの収束が遅いときはそのままでは実用にならないが，加速法を適用することにより利用可能となることがある．

本報告では加速法の概要，研究の現状について紹介する．まず数列を扱い，続いてベクトル列を扱う．

2. 数列の収束の速さ

数列 (s_n) が s に p 次収束するとは，

$$\lim_{n\to\infty}\frac{|s_{n+1}-s|}{|s_n-s|^p}=C,\quad (C>0)$$

のときいう．また，s に収束する数列 (s_n) が極限値

$$\lim_{n\to\infty}\frac{s_{n+1}-s}{s_n-s}=\lambda$$

を持つときは $|\lambda|\le 1$ となる．$\lambda=0$ のとき超線型収束，$-1\le\lambda<1$, $\lambda\ne 0$ のとき線型収束，$\lambda=1$ のとき対数収束という．

一般に，線型収束数列は $|\lambda|$ が 0 に近いときは速く，1 に近いときは遅くなる．対数収束数列の収束の速さは，s_n-s を $n\to\infty$ に関し漸近展開したときの主要項により決まるので，対数収束数列の分類は主要項のオーダーにより行う．たとえば，Riemann ゼータ関数 $\zeta(\alpha)=\sum_{i=1}^{\infty}i^{-\alpha}$の第 n 部分和を s_nとしたとき，

$$s_n-\zeta(\alpha)\sim\frac{1}{1-\alpha}n^{1-\alpha}+\frac{1}{2}n^{-\alpha}+\cdots$$

だから，数列 (s_n) は$\zeta(\alpha)$ に $O(n^{1-\alpha})$ の対数収束をするという．

非線型方程式の反復解法を例にとり，収束の速さを見てみよう．例 1 から例 3 は $x^2-1=0$ を初期値 $s_0=2$ として，それぞれ Newton 法，割線法 $(s_1=1.25)$ および簡易 Newton 法

で求めたもので，例 4 と例 5 は $(x-1)^2=0$ を初期値 $s_0=2$ として，Newton 法および簡易 Newton 法で求めたものである．

例 1
$$\begin{cases} s_0 = 2 \\ s_{n+1} = \dfrac{1}{2}\left(s_n + \dfrac{1}{s_n}\right) \quad n = 0, 1, \ldots \end{cases}$$

例 2
$$\begin{cases} s_0 = 2 \\ s_1 = 1.25 \\ s_{n+1} = s_n - \dfrac{s_n^2 - 1}{s_n + s_{n-1}} \quad n = 1, 2, \ldots \end{cases}$$

例 3
$$\begin{cases} s_0 = 2 \\ s_{n+1} = s_n - \dfrac{1}{4}\left(s_n^2 - 1\right) \quad n = 0, 1, \ldots \end{cases}$$

例 4
$$\begin{cases} s_0 = 2 \\ s_{n+1} = \dfrac{1}{2}\left(s_n + 1\right) \quad n = 0, 1, \ldots \end{cases}$$

例 5
$$\begin{cases} s_0 = 2 \\ s_{n+1} = s_n - \dfrac{1}{2}\left(s_n - 1\right)^2 \quad n = 0, 1, \ldots \end{cases}$$

計算結果は表 1 のようになる．小数第 11 位を 4 捨 5 入して表した．

表 1

n	例 1	例 2	例 3	例 4	例 5
1	1.25	1.25	1.25	1.5	1.5
2	1.025	1.07692 30769	1.10937 5	1.25	1.375
3	1.00030 48780	1.00826 44628	1.05169 67773	1.125	1.30468 75
4	1.00000 00465	1.00030 48780	1.02518 02495	1.0625	1.25827 02637
5	1.00000 00000	1.00000 12545	1.01243 16135	1.03125	1.22491 84991
6	1.00000 00000	1.00000 00002	1.00617 71705	1.01562 5	1.19962 43335
7	1.00000 00000	1.00000 00000	1.00307 90459	1.00781 25	1.17969 93962
8	1.00000 00000	1.00000 00000	1.00153 71528	1.00390 625	1.16355 34597
9	1.00000 00000	1.00000 00000	1.00076 79857	1.00195 3125	1.15017 85926
10	1.00000 00000	1.00000 00000	1.00038 38454	1.00097 65625	1.13890 17878

例 1 は 2 次収束，例 2 は $(1+\sqrt{5})/2 = 1.618$ 次収束である．例 3 と例 4 は線型収束で，例 5 は $O(n^{-1})$ の対数収束である．

例 1 は反復 1 回ごとに有効桁数が 2 倍に増える．例 3 と例 4 は反復 1 回ごとに誤差が半分になる．例 5 は収束が極めて遅く，このままでは実用にならない．

3．数列の収束の加速と数列変換

数列の集合から数列の集合への写像を数列変換[1] *sequence transformation* という．数列変換の典型的な例を示そう．数列 (s_n) が

$$s_{n+1} - s \fallingdotseq \lambda(s_n - s), \quad \lambda \neq 1$$

を満たしていると仮定する．$\lambda > 1$ のとき (s_n) は発散する．このとき s は (s_n) の反極限 *anti-limit*(Shanks[28]) という．

λが既知のとき，上式を s について解いた式を t_nとおくと，

$$t_n = \frac{s_{n+1} - \lambda s_n}{1 - \lambda}.$$

数列変換 $T : (s_n) \mapsto (t_n)$ を Richardson 補外という．

λが未知のとき，

$$\frac{s_{n+2} - s}{s_{n+1} - s} \fallingdotseq \frac{s_{n+1} - s}{s_n - s}$$

を s について解いた式を t_nとおくと，

$$\begin{aligned} t_n &= \frac{s_n s_{n+2} - s_{n+1}^2}{s_{n+2} - 2s_{n+1} + s_n} \\ &= s_n - \frac{(s_{n+1} - s_n)^2}{s_{n+2} - 2s_{n+1} + s_n} \\ &= s_n - \frac{(\Delta s_n)^2}{\Delta^2 s_n} \end{aligned}$$

となる．数列変換 $T : (s_n) \mapsto (t_n)$ を Aitken Δ^2法という．t_nは，

$$\begin{aligned} t_n &= s_{n+1} - \frac{(s_{n+2} - s_{n+1})(s_{n+1} - s_n)}{s_{n+2} - 2s_{n+1} + s_n} \\ &= s_{n+2} - \frac{(s_{n+2} - s_{n+1})^2}{s_{n+2} - 2s_{n+1} + s_n} \\ &= s_{n+2} - \frac{(\nabla s_{n+2})^2}{\nabla^2 s_{n+2}} \end{aligned}$$

とも表せる．

[1] 日本語訳が定着していないか筆者が日本語に訳した用語には，日本語と英語の双方を記した．

数列 (s_n) は s に収束するものとする．数列変換 $T:(s_n)\mapsto(t_n)$ が，数列 (s_n) の収束を加速するとは

$$\lim_{n\to\infty}\frac{t_n-s}{s_{\sigma(n)}-s}=0$$

のときいう．ここで，$\sigma(n)$ は t_n が $s_1,\ldots,s_{\sigma(n)}$ により決定されるような最小の自然数である．たとえば，Aitken Δ^2法

$$t_n=s_n-\frac{(s_{n+1}-s_n)^2}{s_{n+2}-2s_{n+1}+s_n}$$

が (s_n) の収束を加速するのは，$\sigma(n)=n+2$ だから

$$\lim_{n\to\infty}\frac{t_n-s}{s_{n+2}-s}=0$$

のときである．よく知られているように，線型収束列は Aitken Δ^2法により加速される．(Henrici[17,p.73])．

先の例 3 に Richardson 補外と Aitken Δ^2法を適用した結果は表 2 のとおりである．

表 2

n	例 3	Richardson 補外	Aitken Δ^2法
1	1.25	0.5	
2	1.10937 5	0.96875	1.07692 30769
3	1.05169 67773	0.99401 85546	1.01158 94039
4	1.02518 02495	0.99866 37216	1.00261 64494
5	1.01243 16135	0.99968 29775	1.00062 67765
6	1.00617 71705	0.99992 27274	1.00015 36264
7	1.00307 90459	0.99998 09212	1.00003 80421
8	1.00153 71528	0.99999 52597	1.00000 94660
9	1.00076 79857	0.99999 88185	1.00000 23610
10	1.00038 38454	0.99999 97050	1.00000 05895

4．加速の起源

加速法として今日最も有名な Richardson 補外と Aitken Δ^2法は，17 世紀から 18 世紀にヨーロッパと日本において円周率の計算の際に用いられたのが，始まりである．

直径 1 の円に内接する正 n 角形の周を T_n とし，外接する正 n 角形の周を U_n とする．

C. Huygens は，1654 年に著書 *De Circuli Magnitudine Inventa* において Euclid 幾何の手法で不等式

$$T_{2n}+\frac{1}{3}(T_{2n}-T_n)<\pi<\frac{2}{3}T_n+\frac{1}{3}U_n \tag{4.1}$$

を証明した．(平山 [18,pp.75-76] に解説がある.)

$$
\begin{aligned}
T_n &= n\sin\frac{\pi}{n}\\
&= \pi\left(1-\frac{1}{3!}(\frac{\pi}{n})^2+\frac{1}{5!}(\frac{\pi}{n})^4-\dots\right)\\
U_n &= n\tan\frac{\pi}{n}\\
&= \pi\left(1+\frac{1}{3}(\frac{\pi}{n})^2+\frac{2}{15}(\frac{\pi}{n})^4+\dots\right)
\end{aligned}
$$

より，(4.1) の左辺は$\lambda = 1/4$ として (T_{2n}) に Richardson 補外を適用したものに一致し，右辺は数値積分の Simpson 則が中点則と台形則の 2:1 の重み付き平均により得られるのと類似の加速法である．

次に，$s_n = T_{2^n}$ とおく．1681~84 年頃，関孝和は

$$
\begin{aligned}
s_{15} &= 3.14159\ 26\mathit{487}\ \mathit{76985}\ \mathit{6708},\\
s_{16} &= 3.14159\ 265\mathit{23}\ \mathit{86591}\ \mathit{3571},\\
s_{17} &= 3.14159\ 2653\mathit{2}\ \mathit{88992}\ \mathit{7759},
\end{aligned}
$$

から

$$
\begin{aligned}
t_{15} &= s_{16}+\frac{(s_{16}-s_{15})(s_{17}-s_{16})}{(s_{16}-s_{15})-(s_{17}-s_{16})},\\
&= 3.14159\ 26535\ 89793\ 2\mathit{476},
\end{aligned}
\tag{4.2}
$$

により円周率を計算した．(藤原 [14,p.180] をみよ)．(4.2) は Aitken Δ^2法である[2]．関の計算は没後 1712 年に弟子の荒木村英により篇纂された括要算法の中で発表された．

関の弟子の建部賢弘は，1722 年に出版した綴術算経において

$$
\begin{aligned}
\sigma_0^{(n)} &= (s_n)^2 \quad n=2,3,\dots,\\
\sigma_k^{(n)} &= \sigma_{k-1}^{(n+1)}+\frac{\sigma_{k-1}^{(n+1)}-\sigma_{k-1}^{(n)}}{2^{2k}-1} \quad n=2,3,\dots;k=1,2,\dots
\end{aligned}
$$

により，円周率を 41 桁計算した．([14,pp.296-298])

$$
\sqrt{\sigma_8^{(2)}} = 3.14159\ 26535\ 89793\ 23846\ 26433\ 83279\ 50288\ 41971\ \mathit{2}
$$

[2] この事実を最初に指摘したのは，C. Brezinski である．

建部の計算は，Richardson 補外の繰り返し適用である．

関の発見はヨーロッパにおける H.Von Nägelsbach による再発見 ([1, p.305]) より 190 年以上，建部の発見は J.F.Saigey による再発見 (Dutka[11]) より 130 年以上早い．

5．加速法研究の現状

数列の加速法の研究は，最近 10 数年の間に大きく様変わりしている．特に，

(i) 数列の集合と数列変換の諸性質

(ii) 行列式の比で与えられる数列変換に対する行列式を用いない効率的算法

などにおいて，著しい結果が得られている．それらの一端を紹介しよう．(i) については，Delahaye[8]，加速法全般については Brezinski and Redivo Zaglia[6] が詳しい．

5.1 数列の集合と数列変換

よく現れる数列の集合には，以下のものがある．

$\mathscr{C}$ =収束する複素数列全体の集合,

$$\mathscr{C}^* = \left\{ (s_n) \in \mathscr{C} \;\middle|\; s_n \neq \lim_{m\to\infty} s_m \ \forall n \right\},$$

$$\mathrm{LIN} = \left\{ (s_n) \in \mathscr{C}^* \;\middle|\; \exists\lambda \neq 0, -1 \leq \lambda < 1,\ \lim_{n\to\infty} \frac{s_{n+1}-s}{s_n - s} = \lambda \right\},$$

$$\mathrm{LOG} = \left\{ (s_n) \in \mathscr{C}^* \;\middle|\; \lim_{n\to\infty} \frac{s_{n+1}-s}{s_n - s} = 1 \right\},$$

$$\mathrm{LOGSF} = \left\{ (s_n) \in \mathrm{LOG} \;\middle|\; \lim_{n\to\infty} \frac{s_{n+2}-s_{n+1}}{s_{n+1} - s_n} = 1 \right\}.$$

数列変換 $T\colon (s_n) \mapsto (t_n)$ に対し幾つかの性質を定義しておく．$\sigma(n) = n + k$とする．

(t_n) が (s_n) と同じ極限値に収束するとき，Tは (s_n) に対し正則 *regular* であるという．

極限値

$$\lim_{n\to\infty} \frac{t_n - s}{s_{n+k} - s} = \lambda, \quad (\lambda \neq 1)$$

が存在するとき，Tは (s_n) に同時的 *synchronous*(Germain-Bonne and Kowalewski[15]) であるという．とくに，$\lambda = 0$ のときは，Tは (s_n) の収束を加速するという．Tが同時的であればλが未知であっても，(s_n) の収束を加速することができる．(Litovsky[21])

自然数 Nが存在し，$t_n = s$, $\forall n > N$ のとき，Tは (s_n) に対し完全 *exact* であるという．

数列変換 $T\colon \mathscr{S} \to \mathscr{T}$が与えられたとき，以下の問題が基本的である．

1. Tが正則に働く$\mathscr{S}$の最大の部分集合を求めよ．この部分集合をTの正則域 *domain of regularity* といい，$\mathscr{R}_T$で表す．
2. Tによって加速される$\mathscr{S}$の最大の部分集合を求めよ．この部分集合をTの加速域 *domain of acceleration* といい，$\mathscr{A}_T$で表す．

3. Tが完全に働く$\mathscr{S}$の最大の部分集合を求めよ．この部分集合をTの核 *kernel* といい，$\mathscr{N}_T$で表す．
4. Tは$\mathscr{S}$のすべての数列を加速するか．このとき，$\mathscr{S}$は加速可能集合 *accelerable set*, Tは普遍数列変換 *universal sequence transformation* という．

例 6　任意の線型収束数列は，Aitken Δ^2 法により加速される (Henrici[17]) ので，LIN は加速可能集合で Aitken Δ^2 法が LIN の普遍数列変換である．　—

例 7　Aitken Δ^2 法

$$T : s_n \longmapsto s_n - \frac{(s_{n+1}-s_n)^2}{s_{n+2}-2s_{n+1}+s_n}$$

については，以下のようになる．

$$\mathscr{S} = \left\{ (s_n) \in \mathscr{C}^* \;\middle|\; s_{n+2}-2s_{n+1}+s_n \neq 0,\ \forall n \right\}$$
$$\mathscr{R} = \left\{ (s_n) \in \mathscr{S} \cap \mathscr{C}^* \;\middle|\; \lim_{n\to\infty} \frac{(s_{n+1}-s_n)^2}{s_{n+2}-2s_{n+1}+s_n} = 0 \right\}$$
$$\mathscr{A} = \left\{ (s_n) \in \mathscr{R} \;\middle|\; \lim_{n\to\infty} \frac{(s_{n+2}-s_{n+1})^2}{(s_{n+2}-s)(s_{n+2}-2s_{n+1}+s_n)} = 1 \right\}$$
$$\mathscr{N} = \left\{ (s_n) \in \mathscr{S} \;\middle|\; \exists\lambda \in \mathbb{C}^*, \lambda \neq 1,\ \ \frac{s_{n+1}-s}{s_n-s} = \lambda \right\} \quad —$$

次に，加速可能でない場合を考える．このための重要な概念として，一般化残留 (*generalized remanence*) がある．

数列の集合$\mathscr{S}$が一般化残留[3](Delahaye and Germain-Bonne[10]) 性を持つとは次の 2 条件を満たすときいう．

a). $\hat{s}$に収束する数列 $(\hat{s}_n) \in \mathscr{S}$ s.t. $\exists N,\ \forall n > N,\ \hat{s}_n \neq \hat{s}$ で以下の条件を満たすものが存在する．

i) $\exists (s_n^0) \in \mathscr{S}$ s.t. $\lim_{n\to\infty} s_n^0 = \hat{s}_0$

ii) $\forall m_0,\ \exists p_0 \geq m_0\quad \exists (s_n^1) \in \mathscr{S}$ s.t. $\lim_{n\to\infty} s_n^1 = \hat{s}_1,\ \forall m \leq p_0,\ s_m^1 = s_m^0$

iii) $\forall m_1 > p_0,\ \exists p_1 \geq m_1,\ \exists (s_n^2) \in \mathscr{S}$ s.t. $\lim_{n\to\infty} s_n^2 = \hat{s}_2$
$\forall m \leq p_1,\ s_m^2 = s_m^1$

iv) $\ldots$

b). $(s_0^0, \ldots, s_{p_0}^0, s_{p_0+1}^1, \ldots, s_{p_1}^1, s_{p_1+1}^2, \ldots, s_{p_2}^2, s_{p_2+1}^3, \ldots) \in \mathscr{S}$.

以上の準備のもとで，加速法の基本定理を与える．

定理 1 (Delahaye and Germain-Bonne[10])

数列の集合$\mathscr{S}$が一般化残留性を満たすならば，一つの数列変換ですべての$\mathscr{S}$の数列を加速するものは存在しない．(i.e. $\mathscr{S}$は加速不可能である.)

[3]残留とは，b) の数列が$\mathscr{S}$に残留していることから名付けられた．フランス語の rémanence からきている．

定理 2 (Delahaye and Germain-Bonne[9])
Eを距離空間とし，$\mathscr{C}^*$をEの収束列の集合で

$$\mathscr{C}^* = \{\, (s_n) \mid s_n \in E,\ s_n \neq s \text{ for } \forall n \in \mathbb{N}\}$$

とする．$\mathscr{C}^*$が加速可能であるための必要十分条件は，E''が空集合である．ここで，E'はEの集積点の集合で$E'' = (E')'$を表す．

定理 3 (Delahaye and Germain-Bonne[9])
すべての対数収束数列を加速する数列変換は存在しない．(i.e. LOG は加速不可能.)

定理 4 (Delahaye and Germain-Bonne[10])
LOGSF は加速不可能.

定理 4 より，対数収束数列の加速については，以下のことがいえる．

(i) LOGSF は大きすぎるので，部分集合を考える必要がある．(Delahaye and Germain-Bonne[10])

(ii) 加速可能な部分集合で比較的大きなものに

$$\mathscr{L} = \left\{\, (s_n) \in \mathrm{LOGSF} \;\middle|\; \exists a \in \mathbb{R}^*,\ \lim_{n\to\infty} \frac{1-(s_{n+1}-s)/(s_n-s)}{1-\Delta s_{n+1}/\Delta s_n} = a \,\right\}$$

がある[4] (Brezinski[5,p.398]).

(iii) 実用上は，ある型の漸近展開を満たす数列の集合と，そのような数列を加速する数列変換を考える．

(iv) 数列 (s_n) が漸近展開

$$s_n \sim s + \sum_{j=1}^{\infty} c_j g_j(n)$$

を満たすとき (t_n) の満たす漸近公式，たとえば

$$t_n = s + O(h_1(n))$$
$$t_n \sim s + \sum_{j=1}^{\infty} d_j h_j(n)$$

を与える．

[4] 0 以外の極限値が存在するというだけで，a は必ずしも計算できなくてもよい．Aitken Δ^2法により (s_n) は同時的になる．

5.2 漸近展開に基づく加速法

数列 (s_n) が漸近展開

$$s_n - s \sim \sum_{j=1}^{\infty} c_j g_j(n)$$

されるとする．ここで，c_jは未知の定数で，$g_j(n)$ は既知の n の関数とする．このとき $s, c_1, \ldots, c_k$に関する連立 1 次方程式

$$\begin{cases} s_n &= s + c_1 g_1(n) + \cdots + c_k g_k(n) \\ s_{n+1} &= s + c_1 g_1(n+1) + \cdots + c_k g_k(n+1) \\ & \cdots \\ s_{n+k} &= s + c_1 g_1(n+k) + \cdots + c_k g_k(n+k) \end{cases}$$

を Cramer の公式により s について解いた式を $T_k^{(n)}$ とおくと，

$$T_k^{(n)} = \frac{\begin{vmatrix} s_n & s_{n+1} & \cdots & s_{n+k} \\ g_1(n) & g_1(n+1) & \cdots & g_1(n+k) \\ & & \cdots & \\ g_k(n) & g_k(n+1) & \cdots & g_k(n+k) \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ g_1(n) & g_1(n+1) & \cdots & g_1(n+k) \\ & & \cdots & \\ g_k(n) & g_k(n+1) & \cdots & g_k(n+k) \end{vmatrix}}$$

となる．数列変換 $T : (s_n) \mapsto (T_k^{(n)})$ は (s_n) の収束を加速することが期待される．このような行列式の商として表される数列変換には以下のようなものがある．

(i) Aitken Δ^2 法：$k = 1$，$g_1(n) = \Delta s_n$.

(ii) Richardson 補外：$g_j(n) = (4^{-j})^n = (4^{-n})^j$.

(iii) Shanks e-変換 [28]：$g_j(n) = \Delta s_{n+j-1}$.

(iv) Levin u-変換 [19]：$g_j(n) = n^{1-j} \Delta s_{n-1}$.

(v) Levin and Sidi[20] $d^{(m)}$-変換：$g_{pm+q}(n) = n^{q-p} \Delta^q s_{n-q}$.

行列式の数値計算は一般には手間が掛かるが，上記の場合は分母と分子の規則性のため漸化式により効率よく計算できる．1975 年に C. Schneider[27]，1979 年に T. Håvie[16]，1980 年に C. Brezinski[4] が独立に次のような算法を与えた．

2 次元配列 $E_k^{(n)}$ と 3 次元配列 $g_{k,j}^{(n)}$ を次のように定義する．

$$E_0^{(n)} = s_n, \quad n = 0, 1, \ldots,$$
$$g_{0,i}^{(n)} = g_i(n), \quad n = 0, 1, \ldots; i = 1, 2, \ldots$$

$k=1,2,\ldots;\ n=0,1,\ldots$ に対し

$$E_k^{(n)}=E_{k-1}^{(n)}-\frac{E_{k-1}^{(n+1)}-E_{k-1}^{(n)}}{g_{k-1,k}^{(n+1)}-g_{k-1,k}^{(n)}}\cdot g_{k-1,k}^{(n)},$$

$$g_{k,i}^{(n)}=g_{k-1,i}^{(n)}-\frac{g_{k-1,i}^{(n+1)}-g_{k-1,i}^{(n)}}{g_{k-1,k}^{(n+1)}-g_{k-1,k}^{(n)}}\cdot g_{k-1,k}^{(n)},\quad i=k+1,k+2,\ldots$$

このとき，$E_k^{(n)}=T_k^{(n)}$ となる．この算法を Brezinski[80] は，E-算法と名付けた．さらに，1987 年には Ford and Sidi[13] が，E-算法よりも効率の良い算法を提案している．

6．ベクトル列の加速

ベクトル列の収束についても，数列に類似のことがらが成立する．絶対値をノルムに替えるなどの方法により，数列に関する定義をベクトル列に拡張できる．

6.1 ベクトル列の収束の速さと加速

ベクトル列 (x^n) が x^* に p 次収束するとは，

$$\lim_{n\to\infty}\frac{\|x^{n+1}-x^*\|_\infty}{\|x^n-x^*\|_\infty^p}=C,\quad (C>0)$$

のときいう．ここで，$\|x\|_\infty$ は最大値ノルムを表す．とくに，1 次収束する複素ベクトル列が極限値

$$\lim_{n\to\infty}\frac{\|x^{n+1}-x^*\|_\infty}{\|x^n-x^*\|_\infty}=\lambda$$

を持つとき，$0<\lambda<1$ ならば線型収束，$\lambda=1$ ならば劣線型収束 *sublinear convergence*(Ortega and Rheinboldt[24,p.286]) という．

x^* に収束する複素ベクトル列 (x^n) が極限値

$$\lim_{n\to\infty}\frac{\langle x^{n+1}-x^*\rangle}{\langle x^n-x^*\rangle}=1$$

を持つとき対数収束という．ここで，$\langle x\rangle$ は，ベクトル $x=(x_1,\ldots,x_N)\in\mathbb{C}^N$ の成分で絶対値が最大のものを示す．すなわち，$\langle x\rangle=x_i$ とは，

(i) $|x_i|=\max\{\,|x_1|,\ldots,|x_N|\,\}=\|x\|_\infty$

(ii) $i=\min\{\,j\mid |x_j|=\|x\|_\infty\}$

のとき．(Osada[26,p.122])

ベクトル列変換 $T:(x^n)\mapsto(y^n)$ がベクトル列 (x^n) の収束を加速するとは

$$\lim_{n\to\infty}\frac{\|y^n-x^*\|}{\|x^{\sigma(n)}-x^*\|}=0$$

のときいう．ここで，$\sigma(n)$ は y^nが $x^1,\ldots,x^{\sigma(n)}$により決定されるような最小の自然数を表す．

6.2 多項式に基づくベクトル列の加速法

ベクトル列 (s^n) が行列 A とベクトル b により

$$s^{n+1}=As^n+b \tag{6.1}$$

で生成されるとき，ベクトル s^1-s^0に関する行列 A の最小多項式を

$$P(x)=\sum_{i=0}^{k}c_ix^i,\quad c_k=1,\qquad P(A)(s^1-s^0)=0$$

とする．$\Delta s^i=s^{i+1}-s^i$とおくと，

$$c_0\Delta s^0+\cdots+c_{k-1}\Delta s^{k-1}+\Delta s^k=0 \tag{6.2}$$

となる．(6.2) より $c_0,\ldots,c_{k-1}$が決定できれば，s^nの極限 s は，

$$s=(\sum_{i=0}^{k}c_is^{n+i})/\sum_{i=0}^{k}c_i,\quad c_k=1$$

と表せる．(Smith, Ford and Sidi[30])

ベクトル列 (s^n) が，近似的に (6.1) で生成されるときでも，(6.2) の左辺の二乗ノルム

$$\|c_0\Delta s^0+\cdots+c_{k-1}\Delta s^{k-1}+\Delta s^k\|_2 \tag{6.3}$$

を最小にする $c_0,\ldots,c_{k-1}$が決定できれば，s^nの極限 s は

$$s_{n,k}=(\sum_{i=0}^{k}c_is^{n+i})/\sum_{i=0}^{k}c_i \tag{6.4}$$

で近似できる．

(6.3) の最小二乗解は，正規方程式

$$\begin{cases} c_0 u_{n,n} + \cdots + c_{k-1} u_{n,n+k-1} & = -u_{n,n+k} \\ \qquad \cdots & \\ c_0 u_{n+k-1,n} + \cdots + c_{k-1} u_{n+k-1,n+k-1} & = -u_{n+k-1,n+k} \end{cases}$$

を解くことにより得られる．ここで，$u_{i,j} = (\Delta s^i, \Delta s^j)$(内積) である．

正規方程式の解を Cramer の公式で表し，(6.4) に代入し整理すると，

$$s_{n,k} = \frac{\begin{vmatrix} s^n & s^{n+1} & \cdots & s^{n+k} \\ u_{n,n} & u_{n,n+1} & \cdots & u_{n,n+k} \\ & & \cdots & \\ u_{n+k-1,n} & u_{n+k-1,n+1} & \cdots & u_{n+k-1,n+k} \end{vmatrix}}{\begin{vmatrix} 1 & 1 & \cdots & 1 \\ u_{n,n} & u_{n,n+1} & \cdots & u_{n,n+k} \\ & & \cdots & \\ u_{n+k-1,n} & u_{n+k-1,n+1} & \cdots & u_{n+k-1,n+k} \end{vmatrix}}$$

となる．ベクトル列変換 $T : (s^n) \mapsto (s_{n,k})$ を最小多項式補外 (MPE) とよぶ．MPE は，Cabay and Jackson[7] が 1976 年に提案し，1980 年に Skelboe[29] が命名した方法である．

MPE 以外にも，Brezinski[3] が 1975 年に提案した変形最小多項式補外 (MMPE)，1977 年と 1979 年に Mešina[23] と Eddy[12] が独立に提案した被約階数補外 (RRE) 等の多項式に基づくベクトル列の補外法がある．

6.3 数列変換のベクトル列変換への拡張

数列の収束の加速に対しては，多くの数列変換が提案されてきた．これらの幾つかは，何らかの方法によりベクトル列変換に拡張できる．最も簡単な方法は成分ごとに変換を施すことであるが，ベクトルの次数が大きくなると計算回数が増える．このため，ベクトルとしての性質を用いた拡張が考えられている．

初めに，ベクトル空間に関する復習をしておく．Kを実または複素数体とし，EをK上のベクトル空間とする．写像 $f : E \times E \to K$が，対称双線型形式であるとは，次の 3 条件

$$\begin{aligned} f(x+y,z) &= f(x,z) + f(y,z), \\ f(\alpha x, y) &= \alpha f(x,y), \quad \alpha \in K \\ f(x,y) &= f(y,x) \end{aligned} \tag{6.5}$$

を満たすときいう．(6.5) のかわりに

$$f(x,y) = \overline{f(y,x)}$$

となるとき，写像 $f: E\times E\to K$はエルミット形式であるという．ここで，$\overline{\alpha}$は$\alpha\in\mathbb{C}$の共役複素数を表す．エルミット形式 fが正定値であるとは，

$$
\begin{aligned}
&f(x,x)\geq 0, \quad \forall x\in E\\
&f(x,x)=0 \quad \text{となるのは } x=0 \text{ に限る}
\end{aligned}
$$

ときいう．

正定値エルミット形式は内積と呼ばれる．本報告では対称双線型形式を $\langle x,y\rangle$ で表し，内積は (x,y) で表す．

Eを内積 $(\cdot,\cdot)$ を持つベクトル空間としたとき，$x\in E$ $(x\neq 0)$ の逆ベクトルを

$$
x^{-1}=\frac{\bar{x}}{(x,x)}=\frac{\overline{x}}{\|x\|^2}
$$

により定義する．

$E=\mathbb{R}^N$または $E=\mathbb{C}^N$のとき，$x=(x_1,\ldots,x_N)$ と $y=(y_1,\ldots,y_N)$ に対し，

$$
\begin{aligned}
\langle x,y\rangle&=\sum_{i=1}^{N}x_iy_i\\
(x,y)&=\sum_{i=1}^{N}x_i\overline{y_i}
\end{aligned}
$$

はそれぞれ対称双線型形式と内積になる．

ϵ算法を例にとり，ベクトル列への拡張について見てみよう．1955 年に D. Shanks[28] が行列式の比により与えた数列変換を e 変換と名付けた．1956 年に P. Wynn[31] が e 変換に対する漸化式による算法を提案し，ϵ算法と名付けた：数列 (s_n) に対し

$$
\begin{aligned}
&\epsilon_{-1}^{(n)}=0, \quad \epsilon_0^{(n)}=s_n, \quad n=0,1,\ldots,\\
&\epsilon_{k+1}^{(n)}=\epsilon_{k-1}^{(n+1)}+\frac{1}{\epsilon_k^{(n+1)}-\epsilon_k^{(n)}} \quad k,n=0,1,\ldots.
\end{aligned}
$$

P. Wynn[33] は 1962 年に逆ベクトルを用いてϵ算法をベクトル列に拡張した．これがベクトルϵ算法 (VEA) である：ベクトル列 (s^n) に対し

$$
\begin{aligned}
&\epsilon_{-1}^{(n)}=0, \quad \epsilon_0^{(n)}=s^n, \quad n=0,1,\ldots,\\
&\epsilon_{k+1}^{(n)}=\epsilon_{k-1}^{(n+1)}+\frac{\overline{\epsilon_k^{(n+1)}-\epsilon_k^{(n)}}}{\left(\epsilon_k^{(n+1)}-\epsilon_k^{(n)},\epsilon_k^{(n+1)}-\epsilon_k^{(n)}\right)} \quad k,n=0,1,\ldots.
\end{aligned}
$$

1975 年に C. Brezinski[3] は，対称双線型形式を用いてϵ算法をベクトル列に拡張し，位相的ϵ算法[5] と名付けた：ベクトル列 (s^n) に対し

$$\epsilon_{-1}^{(n)} = 0, \quad \epsilon_0^{(n)} = s^n, \quad n = 0, 1, \ldots,$$
$$\epsilon_{2k+1}^{(n)} = \epsilon_{2k-1}^{(n+1)} + \frac{y}{\left\langle y, \epsilon_{2k}^{(n+1)} - \epsilon_{2k}^{(n)} \right\rangle} \quad k, n = 0, 1, \ldots,$$
$$\epsilon_{2k+2}^{(n)} = \epsilon_{2k}^{(n+1)} + \frac{\epsilon_{2k}^{(n+1)} - \epsilon_{2k}^{(n)}}{\left\langle \epsilon_{2k+1}^{(n+1)} - \epsilon_{2k+1}^{(n)}, \epsilon_{2k}^{(n+1)} - \epsilon_{2k}^{(n)} \right\rangle} \quad k, n = 0, 1, \ldots.$$

ここで，yは分母が零とならない任意のベクトルである．

ϵ算法をベクトル列に拡張することと類似の方法により，Lubkin[22] の W算法，Brezinski[2] のθ算法，Wynn[32] のρ算法などがベクトル列に拡張されている．積と逆を表 3 のようにとればよい．VEA は Wynn[33]，TEA と GHT は Brezinski[3]，VWT，EWT，VTH，VTH，TRA は Osada[25] による．算法の詳細は Brezinski and Redivo Zaglia[6] を見よ．

表 3 数列変換のベクトル列変換への拡張

	数列変換	ベクトル列変換	
積	$a \cdot b$	(a, b)	$\langle a, b \rangle$
逆	$\frac{1}{a}$	$\frac{\bar{a}}{(a, a)}$	$\frac{y}{\langle y, a \rangle}$
ϵ算法	Wynn	VEA	TEA
W変換	Lubkin	VWT	EWT
θ算法	Brezinski	VTH	GTH
ρ算法	Wynn	VRA	TRA

表 3 以外の方法でベクトル列変換に拡張されている数列変換もある．Aitken Δ^2法は Henrici[17] をはじめ何人かの研究者が種々の方法で拡張しており（[6, pp.250-252] を見よ)，Levin 変換は Osada[26] が最大絶対値成分 $\langle x \rangle$ を用いて拡張している．

ベクトル列に拡張された変換は，もとの数列変換の持つ性質が伝わるので，W変換や Levin u 変換をベクトル列に拡張したものは，ある種の対数収束するベクトル列に対し有効に働く．[25][26] を見よ．

[5] 命名の由来は，線型位相空間の列について導入したことによる．

6.4. ベクトル列の加速法の現状と展望

数値計算においてしばしば登場する収束の遅いベクトル列には，加速法が役に立つことがある．しかしながら，ベクトル列の加速法は数列の加速法に比べ歴史が浅く，現在のところあまり利用されていない．Richardson 補外が数値積分や常微分方程式において利用されているのとは対象的である．

また，これまでに研究されてきたのは，線型収束するベクトル列に対する加速法が大部分であって，対数収束列あるいは劣線型収束列についてはほとんど研究されていない．

ベクトル列の加速法は，理論面と応用面で今後の発展が期待されている分野であるといえよう．

参考文献

[1] A.C. Aitken, On Bernoulli's numerical solution of algebraic equations, *Proc. Roy. Soc. Edinburgh* Ser. A 46(1926), 289–305.

[2] C. Brezinski, Accélération de suits à convergence logarithmique *C. R. Acad. Sc. Paris* t.273(1971), 727–730.

[3] C. Brezinski, Généralisation de la transformation de Shanks, de la table de Padé et l'ϵ-algorithme, Calcolo, 12(1975), 317–360.

[4] C. Brezinski, A general extrapolation algorithm, *Numer. Math.* 35(1980), 175–187.

[5] C. Brezinski, A new approach to convergence acceleration mehods, in A. Cuyt ed., *Nonlinear Numerical Methods and Rational Approximation*, (D. Reidel Publ., Dordrecht, 1988)

[6] C. Brezinski and M. Redivo Zaglia, *Extrapolation Methods: theory and practice*, (Elsevier, Amsterdam, 1991)

[7] S. Cabay and L. W. Jackson, A polynomial extrapolation method for finding limits and antilimits of vector sequences, *SIAM J. Numer. Anal.* 13(1976), 734–752.

[8] J. P. Delahaye, *Sequence Transformations*, (Springer, Berlin, 1988)

[9] J. P. Delahaye and B. Germain-Bonne, Résultats négatifs en acceleration de la convergence, *Numer. Math.*, 35(1980), 443–457.

[10] J. P. Delahaye and B. Germain-Bonne, The set of logarithmically convergent sequences cannot be accelerated, *SIAM. J. Numer. Anal.*, 19(1982), 840–844.

[11] J. Dutka, Richardson extrapolation and Romberg integration, *Historia Mathematica* 11(1984), 3-21.

[12] R. P. Eddy, Extrapolation to the limit of a vector sequence, in P. C. C. Wang, ed., *Information Linkage Between Applied Mathematics and Industry*, (New York, 1979), 387–396.

[13] W. F. Ford and A. Sidi, An algorithm for a generalization of the Richardson extrapolation process, *SIAM J. Numer. Anal.* 24(1987), 1212-1232.

[14] 藤原松三郎，日本学士院篇，明治前日本数学史第二巻，(岩波書店，1956)

[15] B. Germain-Bonne and C. Kowalewski, Accélération de la convergence par utilisation d'une transformation auxiliaire, *Publication ANO* 77, Université de Lille I, 1982.

[16] T. Håvie, Generalized Neville type extrapolation schemes, *BIT* 19(1979), 204–213.

[17] P. Henrici, *Elements of Numerical Analysis*, (John Wiley, New York, 1964)

[18] 平山諦，円周率の歴史，(大阪教育図書，1980)

[19] D. Levin, Development of non-linear transformations for improving convergence of sequences, *Intern. J. Computer Math.* 3(1973), 371–388.

[20] D. Levin and A. Sidi, Two new classes of nonlinear transformations for accelerating the convergence of infinite integrals and series, *Appl. Math. and Comp.* 9(1981), 175–215.

[21] A. M. Litovsky, *Accélération de la convergence des Ensembles Synchrones*, Thèse, Université de Lille I, 1989.

[22] S. Lubkin, A method of summing infinite series, *J. Res. Nat. Bur. Stand.* 48(1952), 228–254.

[23] M. Mešina, Convergence acceleration for the iterative solution of $x = Ax + f$, *Comput. Methods Appl. Mech. Eng*, 10(1977), 165–173.

[24] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, 1970.

[25] N. Osada, Acceleration methods for vector sequences, *J. Comput. Appl. Math.*, 38(1991), 361–371.

[26] N. Osada, Extensions of Levin's transformations to vector sequences, *Numer. Algorithms*, 2(1992), 121–132.

[27] C. Schneider, Vereinfachte Rekursionen zur Richardson-Extrapolation in Spezialfällen, *Numer. Math.* 24(1975), 177–184.

[28] D. Shanks, Non-linear transformations of divergent and slowly convergent sequences, *J. Math. Phys.* 34(1955), 1–42.

[29] S. Skelboe, Computation of the periodic steady-state response of nonlinear networks by extrapolation methods, *IEEE Trans. Circuits and Systems*, 27(1980), 161–175.

[30] D. A. Smith, W. F. Ford and A. Sidi, Extrapolation methods for vector sequences, *SIAM Rev.*, 29(1987), 199–233.

[31] P. Wynn, On a device for computing the $e_m(S_n)$ transformaion, *MTAC* 10(1956), 91–96.

[32] P. Wynn, On a procrustean technique for the numerical transformaion of slowly convergent sequences and series, *Proc. Camb. Phil. Soc.* 52(1956), 663–671.

[33] P. Wynn, Acceleration techniques for iterated vector and matrix problems, *Math. Comput.*, 16(1962), 301–322.